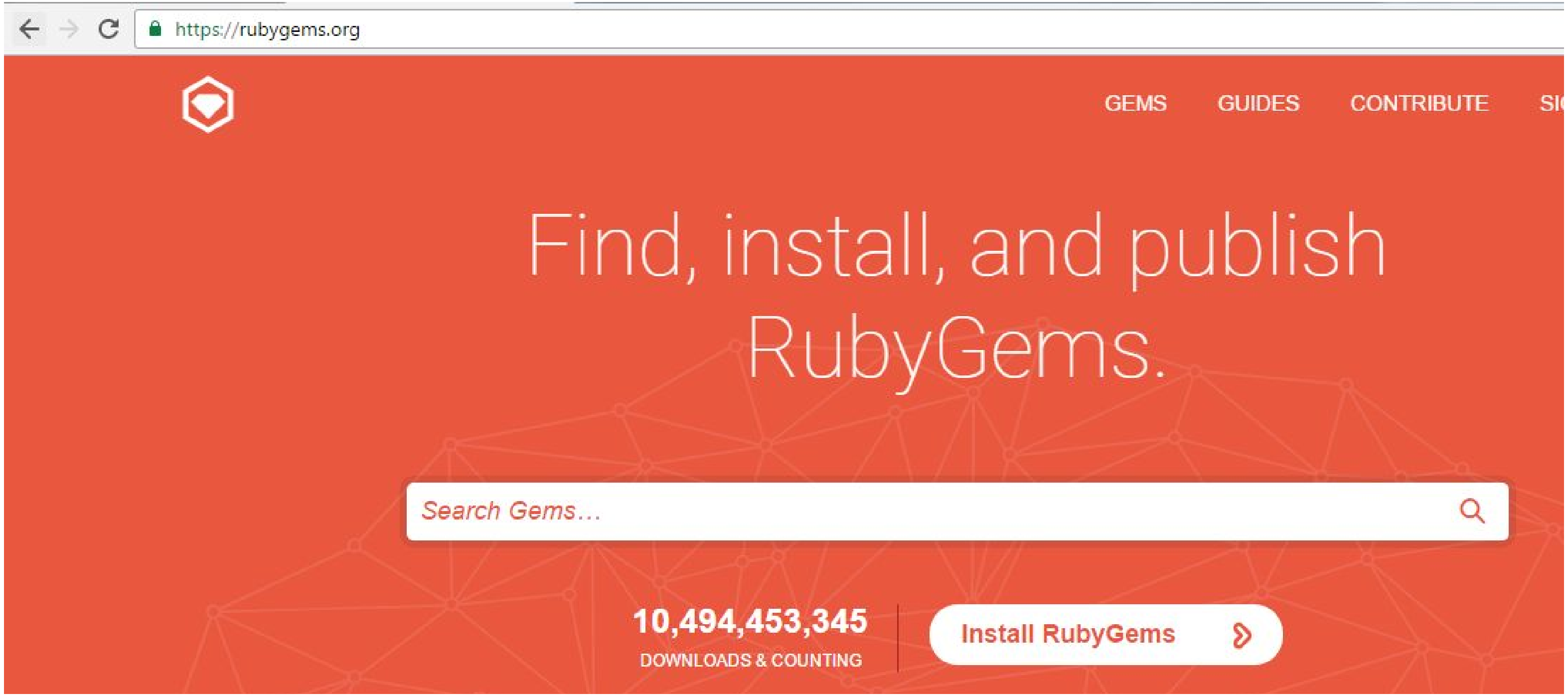
# Introducción a Ruby on Rails

Ruby Gems / HTTP Party

# Overview

- Ruby gems
- Cómo utilizar la gema HTTPParty

# Ruby Gems - 3rd party plugins / additions



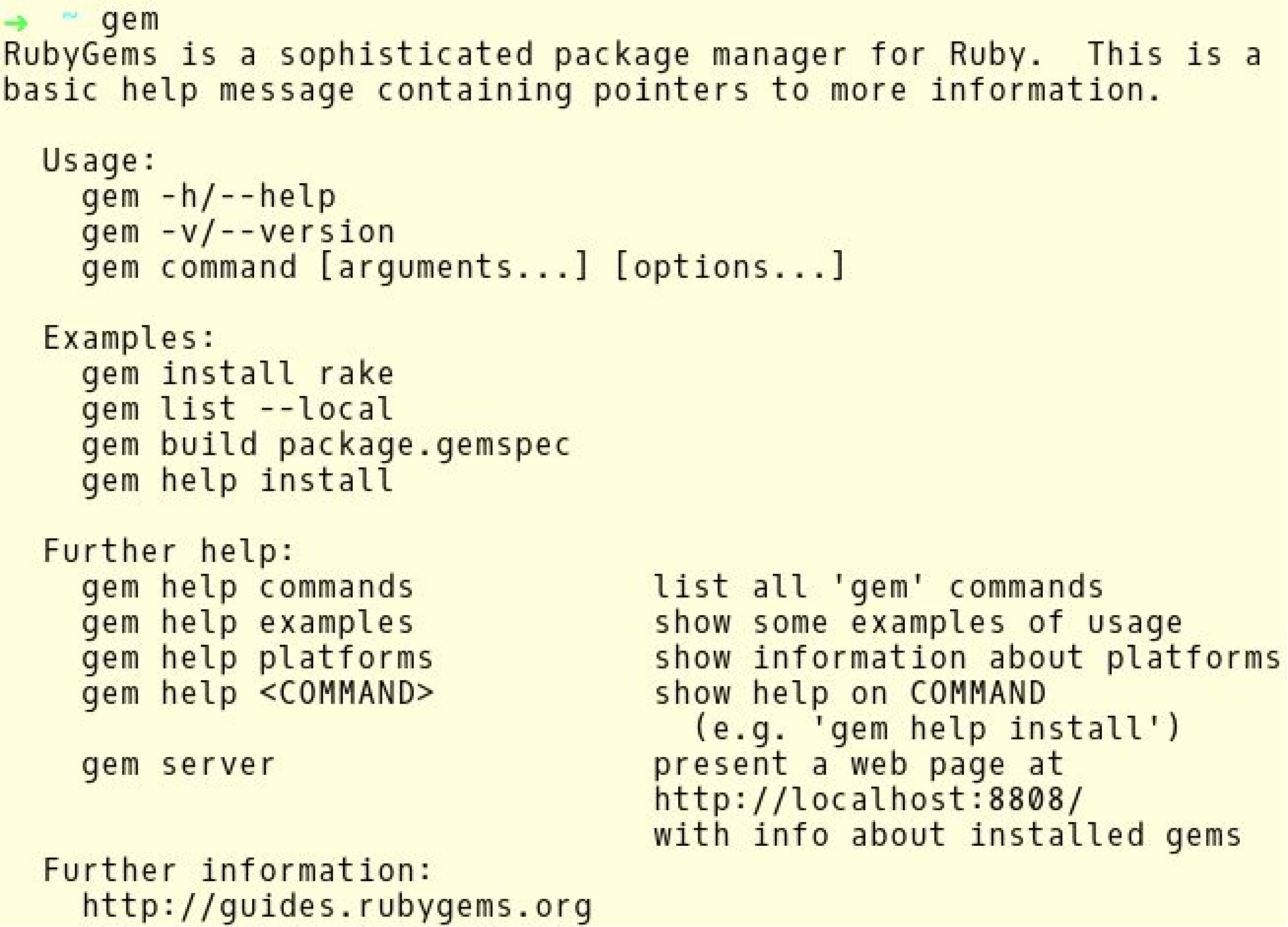
https://rubygems.org

Find, install, and publish RubyGems.

10,494,453,345
DOWNLOADS & COUNTING

# Gem

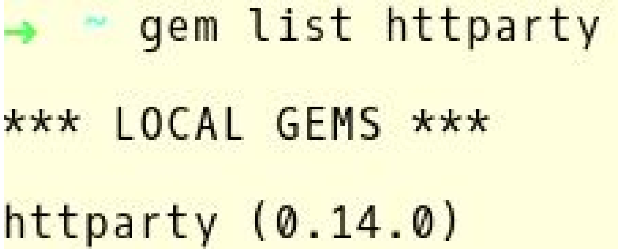
- Package Manager



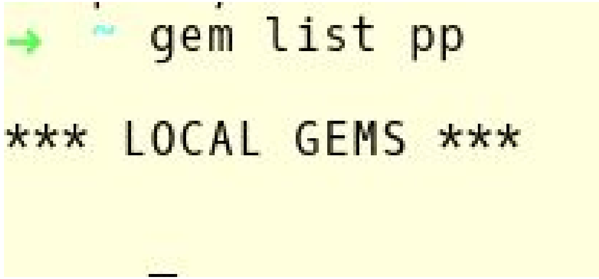
```
➜  ~ gem
RubyGems is a sophisticated package manager for Ruby.  This is a
basic help message containing pointers to more information.

  Usage:
    gem -h/--help
    gem -v/--version
    gem command [arguments...] [options...]

  Examples:
    gem install rake
    gem list --local
    gem build package.gemspec
    gem help install

  Further help:
    gem help commands            list all 'gem' commands
    gem help examples            show some examples of usage
    gem help platforms           show information about platforms
    gem help <COMMAND>           show help on COMMAND
                                   (e.g. 'gem help install')
    gem server                   present a web page at
                                 http://localhost:8808/
                                 with info about installed gems
  Further information:
    http://guides.rubygems.org
```

# RubyGems package manager

- Gem es un package manager que viene instalado con la instalación de ruby.
- **`gem list httpparty`**



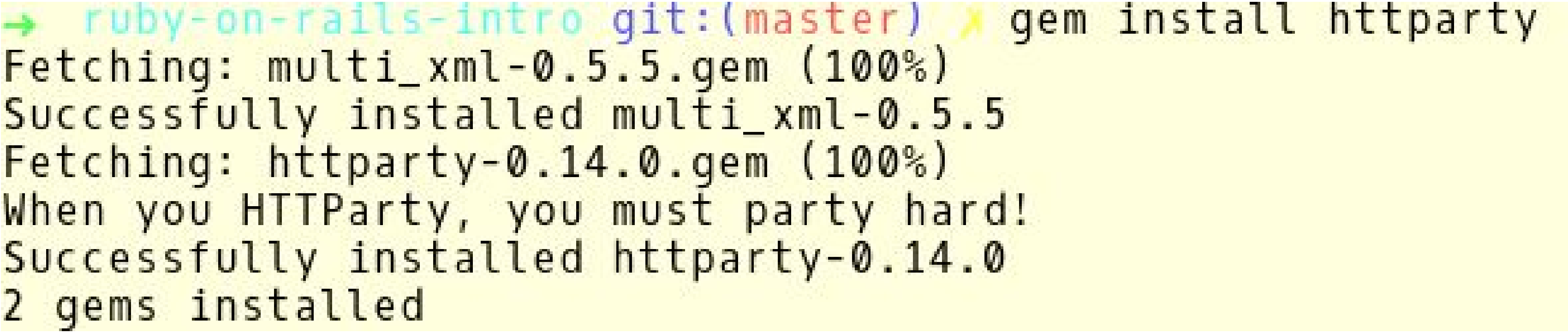
```
→  ~ gem list httparty

*** LOCAL GEMS ***

httparty (0.14.0)
```



```
→  ~ gem list pp

*** LOCAL GEMS ***

```

# Instalando HTTP Party

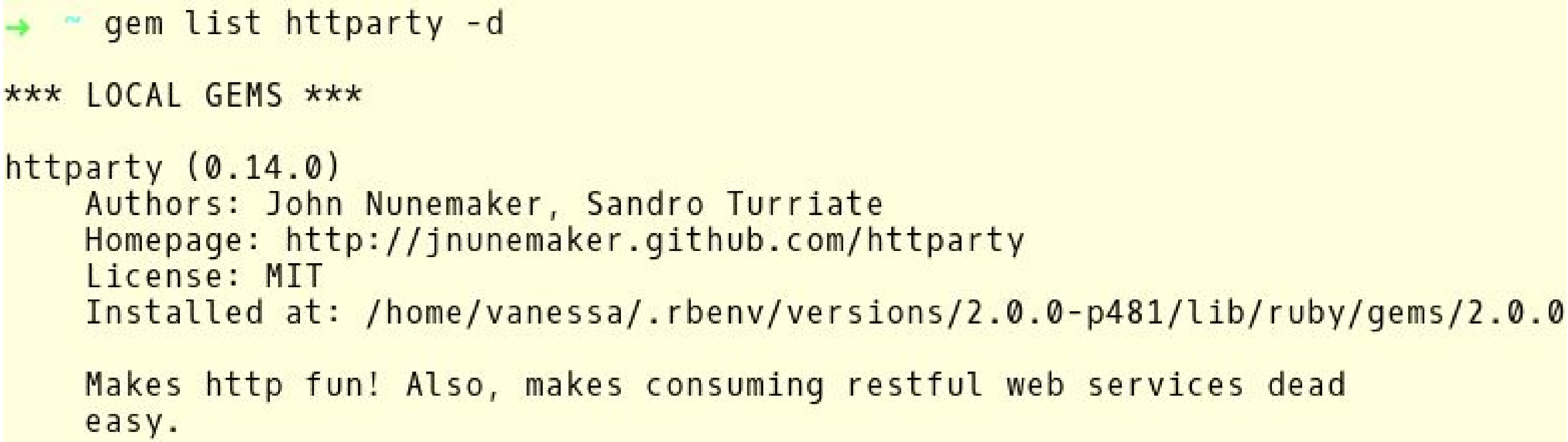
- `gem install httparty`



```
→  ruby-on-rails-intro git:(master) ✗ gem install httparty
Fetching: multi_xml-0.5.5.gem (100%)
Successfully installed multi_xml-0.5.5
Fetching: httparty-0.14.0.gem (100%)
When you HTTParty, you must party hard!
Successfully installed httparty-0.14.0
2 gems installed
```

# Más detalles de los gems

- **gem list httparty -d**



```
→  ~ gem list httparty -d

*** LOCAL GEMS ***

httparty (0.14.0)
    Authors: John Nunemaker, Sandro Turriate
    Homepage: http://jnunemaker.github.com/httparty
    License: MIT
    Installed at: /home/vanessa/.rbenv/versions/2.0.0-p481/lib/ruby/gems/2.0.0

    Makes http fun! Also, makes consuming restful web services dead
    easy.
```

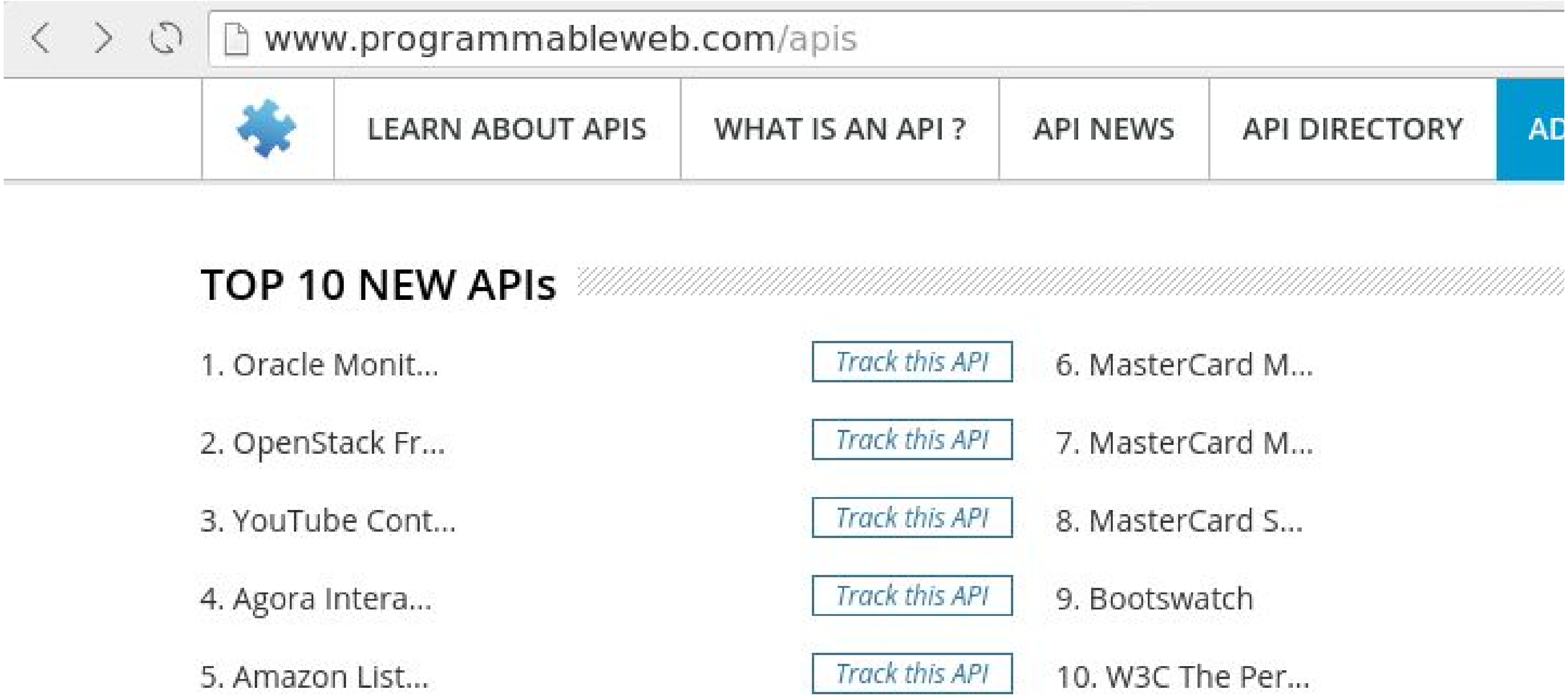
# Qué son los servicios web RestFul?

- Son web services implementados utilizando HTTP y los principios de REST.
  - Tienen una URI base
  - Soportan un formato de intercambio de datos: XML, Json
  - Soportan un set de métodos HTTP (Get, Post, Put, etc).

# HTTParty Gem

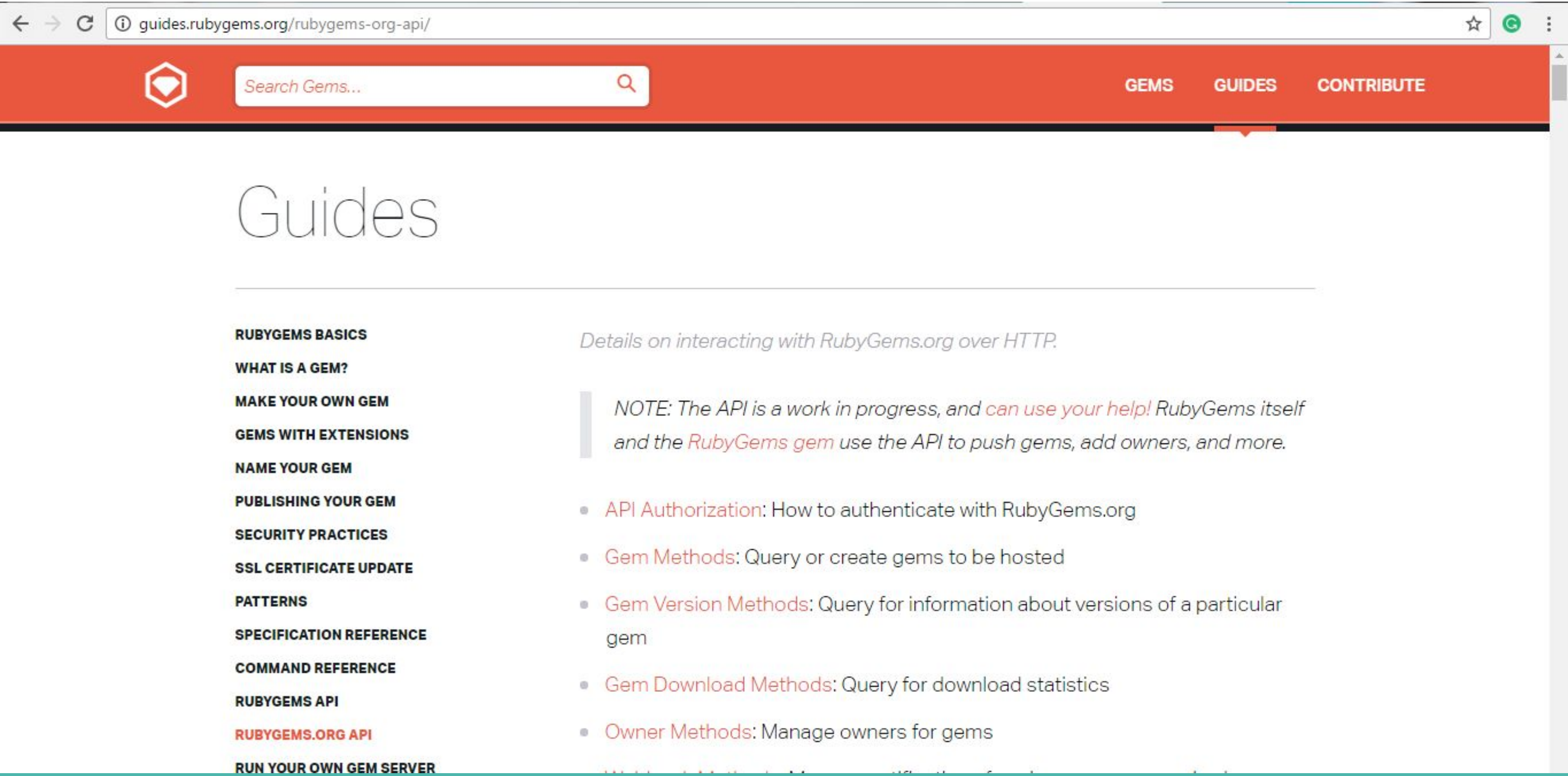
- Cliente de Web Services Restful
    - Comunicación entre máquinas, intercambio de datos.
- Parsing automático de Json y Xml en hashes.
- Provee soporte para:
    - Basic Authentication
    - Parámetros por defecto de los requests

# Existen muchas APIs en todos lados



TOP 10 NEW APIs

1. Oracle Monit...
2. OpenStack Fr...
3. YouTube Cont...
4. Agora Intera...
5. Amazon List...
6. MasterCard M...
7. MasterCard M...
8. MasterCard S...
9. Bootswatch
10. W3C The Per...



# Guides

*Details on interacting with RubyGems.org over HTTP.*

> *NOTE: The API is a work in progress, and can use your help! RubyGems itself and the RubyGems gem use the API to push gems, add owners, and more.*

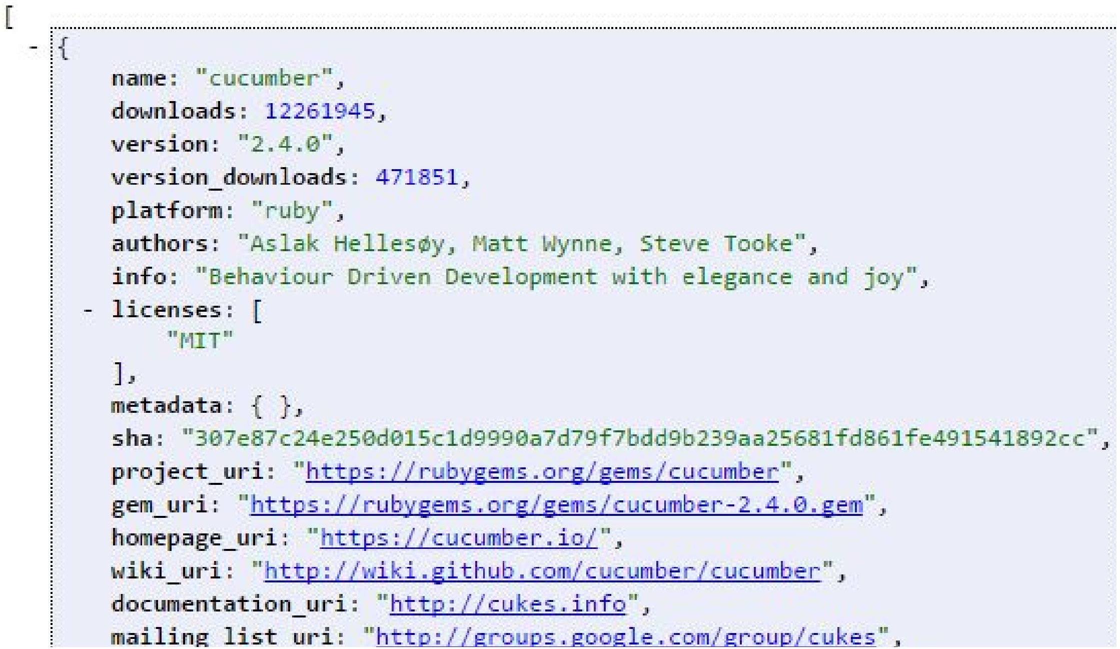
- API Authorization: How to authenticate with RubyGems.org
- Gem Methods: Query or create gems to be hosted
- Gem Version Methods: Query for information about versions of a particular gem
- Gem Download Methods: Query for download statistics
- Owner Methods: Manage owners for gems

# API RUBY GEM

URI base

Parámetros

https://rubygems.org/api/v1/search.json?query=cucumber

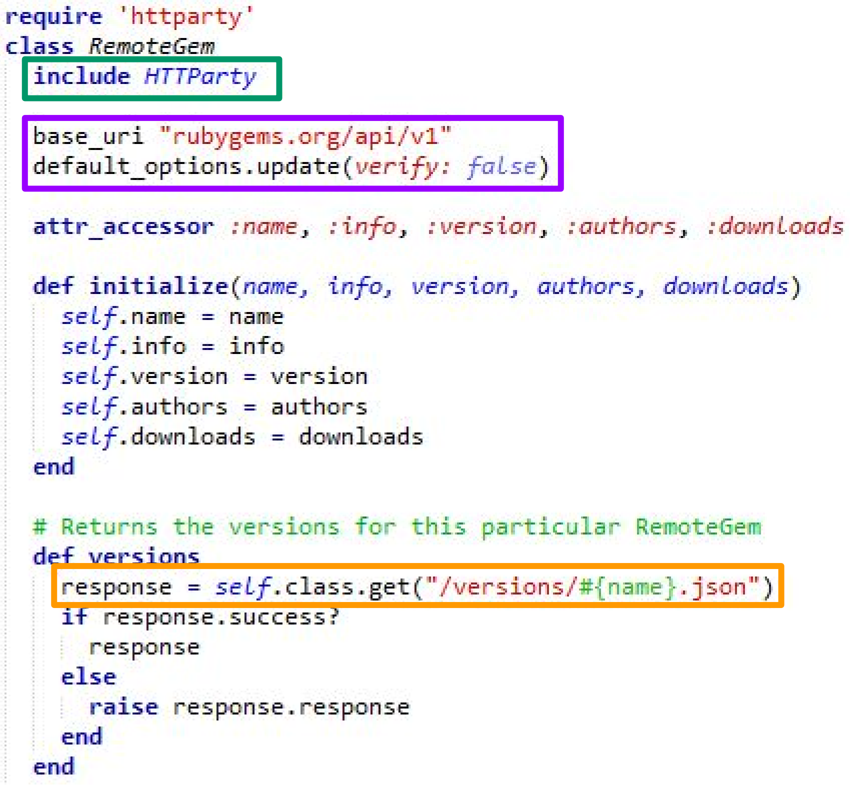


```
[
  - {
        name: "cucumber",
        downloads: 12261945,
        version: "2.4.0",
        version_downloads: 471851,
        platform: "ruby",
        authors: "Aslak Hellesøy, Matt Wynne, Steve Tooke",
        info: "Behaviour Driven Development with elegance and joy",
      - licenses: [
            "MIT"
        ],
        metadata: { },
        sha: "307e87c24e250d015c1d9990a7d79f7bdd9b239aa25681fd861fe491541892cc",
        project_uri: "https://rubygems.org/gems/cucumber",
        gem_uri: "https://rubygems.org/gems/cucumber-2.4.0.gem",
        homepage_uri: "https://cucumber.io/",
        wiki_uri: "http://wiki.github.com/cucumber/cucumber",
        documentation_uri: "http://cukes.info",
        mailing_list_uri: "http://groups.google.com/group/cukes",
```

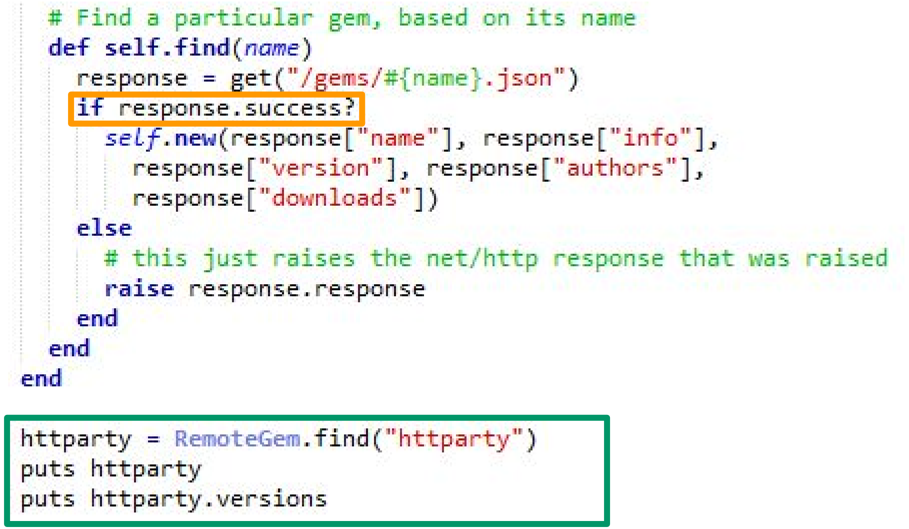
# Como usar HTTParty

- Incluir el módulo
  - include **`HTTParty`**
- Se pueden especificar:
  - **`base_uri`**: para todos los **requests**
  - **`default_params`**: un **API Key** por ejemplo
  - **`format`**: el formato **esperado**

# Test con RubyGems API



```ruby
require 'httparty'
class RemoteGem
  include HTTParty

  base_uri "rubygems.org/api/v1"
  default_options.update(verify: false)

  attr_accessor :name, :info, :version, :authors, :downloads

  def initialize(name, info, version, authors, downloads)
    self.name = name
    self.info = info
    self.version = version
    self.authors = authors
    self.downloads = downloads
  end

  # Returns the versions for this particular RemoteGem
  def versions
    response = self.class.get("/versions/#{name}.json")
    if response.success?
      response
    else
      raise response.response
    end
  end
```



```ruby
  # Find a particular gem, based on its name
  def self.find(name)
    response = get("/gems/#{name}.json")
    if response.success?
      self.new(response["name"], response["info"],
        response["version"], response["authors"],
        response["downloads"])
    else
      # this just raises the net/http response that was raised
      raise response.response
    end
  end
end

httparty = RemoteGem.find("httparty")
puts httparty
puts httparty.versions
```

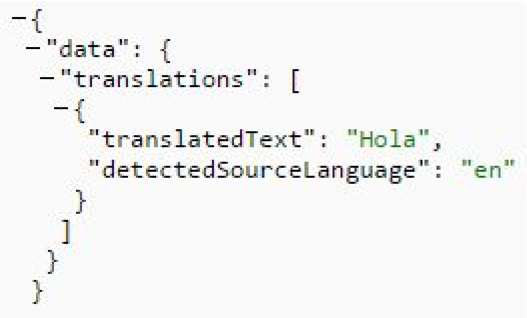
# Practica 1 - Google Translate API (1)

- La API de Google para traducir textos: https://cloud.google.com/translate/v2/translate-reference
- Utilizando Httparty traducir un texto con el servicio:
  - https://www.googleapis.com/language/translate/v2
- Los paramétros:
  - **key** : API KEY habilitado para la utilización de la api, por ahora se puede usar: AIzaSyBdqiJZunlGxiU5ubdmKlp2VqxqR7GYLsQ
  - **q:** el texto que se quiere traducir
  - **source:** idioma de origen (en)
  - **target:** idioma destino (es)

# Practica 1 - Pruebas manuales - Json Respuesta (2)

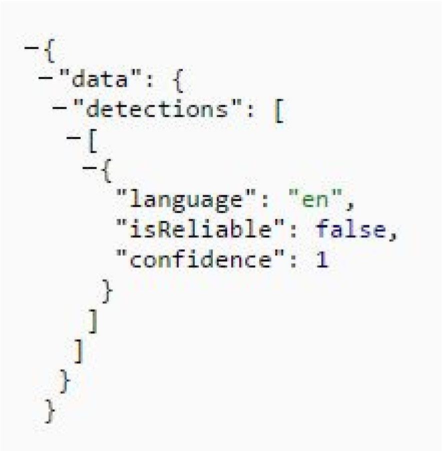
- https://developers.google.com/apis-explorer/?hl=es#p/translate/v2/

## Translate



```
-{
 -"data": {
  -"translations": [
   -{
      "translatedText": "Hola",
      "detectedSourceLanguage": "en"
    }
   ]
  }
 }
```

## Detect



```
-{
 -"data": {
  -"detections": [
   -[
    -{
       "language": "en",
       "isReliable": false,
       "confidence": 1
     }
    ]
   ]
  }
 }
```

# Practica 1 - Google Translate API (2)

- Implementar otro servicio para la detección de lenguaje:
- https://www.googleapis.com/language/translate/v2/detect?parameters
- Parámetros:
  - **key** : API KEY habilitado para la utilización de la api, por ahora se puede usar: AIzaSyBdqiJZunlGxiU5ubdmKlp2VqxqR7GYLsQ
  - **q:** el texto del cual se desea detectar el idioma